Autoencoders and their Applications in Image Processing

Parth Darshan

June 19, 2024

Abstract

This article examines AutoEncoders, which are a specific sort of artificial neural network that aim to acquire effective data representations by reducing dimensionality. We analyze the theoretical basis, structure, and various uses of the subject, which encompass anomaly detection, noise elimination, and Data Compression. The results emphasize the efficacy of AutoEncoders in diverse fields .

1 Introduction

AutoEncoders are a specific kind of artificial neural network that is employed to acquire effective representations of data, usually with the goal of reducing the number of dimensions. These systems operate by compressing the input data into a latent-space representation and subsequently reconstructing the output based on this representation. This procedure is optimized to reduce the disparity between the input and the output. AutoEncoders have a wide range of applications, such as detecting anomalies, removing noise from data, and generating models.

1.1 Outline of the Report

This report is structured as follows:

- **Introduction:** Provides an overview of AutoEncoders and their applications.
- **Exploring AutoEncoders:** Discusses the theoretical foundation of AutoEncoders, including their architecture.
- **Applications:** Details the applications of AutoEncoders.
- **Conclusion:** Summarizes the key points of the report.

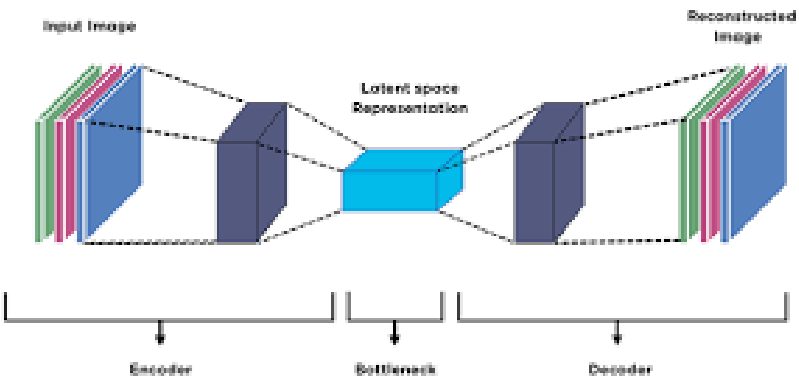


Figure 1: Convolutional AutoEncoder Architecture

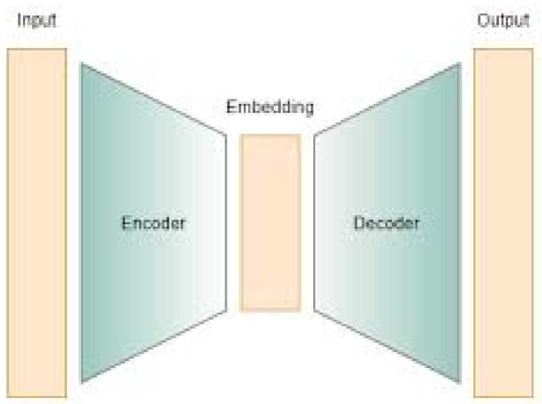
2 The Math behind AutoEncoders

An Autoencoder is a type of neural network that is specifically built to learn an identity function through unsupervised learning. Its purpose is to compress and reconstruct an original input, resulting in a more efficient and compressed representation of the input data.

2.1 Architecture of AutoEncoders

It is important to observe that the AutoEncoder is comprised of two distinct components: the encoder and the decoder.

The encoder component of the AutoEncoder receives the input data and condenses it into a representation in the latent space. AutoEncoders are commonly used for dimensionality reduction because they may create a more compact and concentrated version of the original data. The encoder can be conceptualized as a data compression technique, with the objective of minimizing the data required to represent the initial input.



In contrast, the decoder component of the AutoEncoder utilizes the latent-space representation to regenerate the initial input data. The decoder can be conceptualized as a data decompression algorithm, with the objective of accurately reconstructing the original input from the compressed data.

The AutoEncoder is trained to minimize the discrepancy between the original input and the reconstructed output, which is why the loss function is commonly a mean square error function. This function quantifies the disparity between the initial input and the reconstructed output, and the objective of training is to minimize this disparity.

2.2 Mathematics behind AutoEncoders

AutoEncoders consist of an encoder and a decoder, both of which are neural networks. The encoder maps the input data to a latent space, while the decoder maps the latent space back to the input space. Mathematically, the process involves several steps:

2.2.1 Encoding

The encoder compresses the input data $\mathbf{x} \in R^n$ into a latent representation $\mathbf{z} \in R^m$ where $m < n$. This can be represented as:

$$\mathbf{z} = f(\mathbf{x}) = \sigma(\mathbf{W}_e \mathbf{x} + \mathbf{b}_e)$$

Here,
- $\mathbf{W}_e$ is the weight matrix of the encoder.
- $\mathbf{b}_e$ is the bias vector of the encoder.
- σ is the activation function, typically a non-linear function like ReLU or sigmoid.

2.2.2 Latent Space Representation

The latent space representation $\mathbf{z}$ captures the essential features of the input data in a reduced dimensionality space.

2.2.3 Decoding

The decoder reconstructs the input data from the latent representation. This can be represented as:

$$\hat{\mathbf{x}} = g(\mathbf{z}) = \sigma(\mathbf{W}_d \mathbf{z} + \mathbf{b}_d)$$

Here, - $\mathbf{W}_d$ is the weight matrix of the decoder. - $\mathbf{b}_d$ is the bias vector of the decoder. - σ is the activation function.

2.2.4 Loss Function

The AutoEncoder is trained to minimize the reconstruction error, which is the difference between the input $\mathbf{x}$ and the reconstructed output $\hat{\mathbf{x}}$. The loss function commonly used is the Mean Squared Error (MSE), which is defined as:

$$L(\mathbf{x}, \hat{\mathbf{x}}) = \frac{1}{n} \sum_{i=1}^{n} (x_i - \hat{x}_i)^2$$

2.2.5 Training

The training process involves adjusting the weights $\mathbf{W}_e$, $\mathbf{b}_e$, $\mathbf{W}_d$, and $\mathbf{b}_d$ to minimize the loss function using optimization algorithms like stochastic gradient descent (SGD) or Adam.

2.2.6 Regularization

To prevent overfitting and ensure the model generalizes well to new data, regularization techniques such as L1 or L2 regularization, dropout, or adding noise to the input data are often applied during training.

In summary, the mathematics behind AutoEncoders involves encoding the input data into a lower-dimensional latent space, decoding it back to reconstruct the input, and optimizing the network to minimize the reconstruction error using gradient-based optimization techniques.

3 Applications of AutoEncoders

AutoEncoders are widely used in different fields due to their versatile applicability. In the field of image processing, neural networks are commonly employed to reduce noise in photographs. This is achieved by training the network to eliminate noise from distorted inputs and generate clear images. Anomaly detection involves the use of AutoEncoders to acquire knowledge about the typical patterns in data and discover anomalies by quantifying the reconstruction error, which is typically elevated for atypical data points. Furthermore, AutoEncoders are valuable instruments for compressing data by decreasing the number of dimensions while retaining crucial characteristics, hence facilitating the efficient storage and transmission of extensive datasets.

The following subsections explore a few applications:

3.1 Denoising Images

AutoEncoders are commonly used for the purpose of denoising images. The procedure entails training an AutoEncoder using noisy photos as input and the original clean images as the desired output. The AutoEncoder acquires the ability to recognize and eliminate the noise patterns, thereby reconstructing a clean image from the input that contains noise. The AutoEncoder accomplishes this by leveraging its capacity to extract essential features from the input data while disregarding less significant elements, such as noise. As a result, when a newly received image with noise is inputted into the trained AutoEncoder, it may efficiently diminish the noise and provide a clearer, noise-free image.

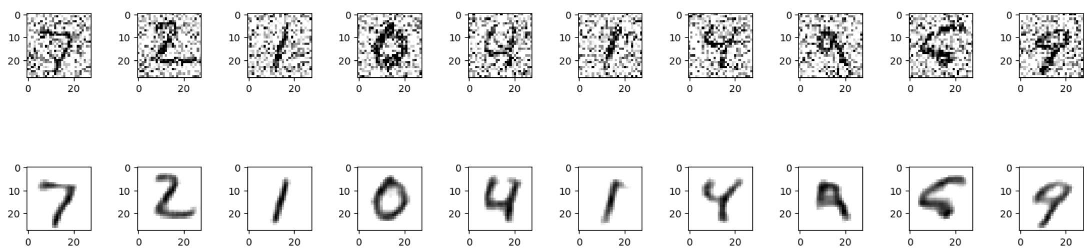


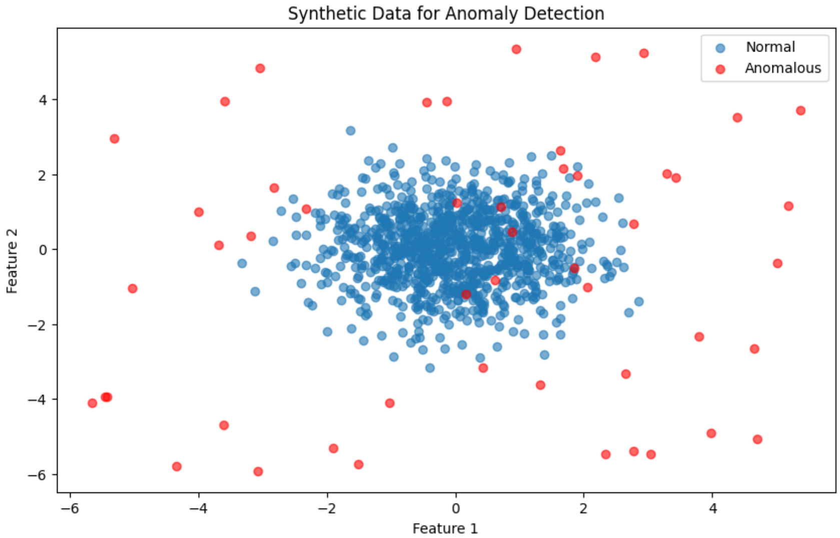
Figure 2: Image Denoising with AutoEncoders

The image above shows the results of denoising when a noisy MNIST dataset was input into the encoder part. In this case, we map the noisy MNIST dataset to the original MNIST dataset.

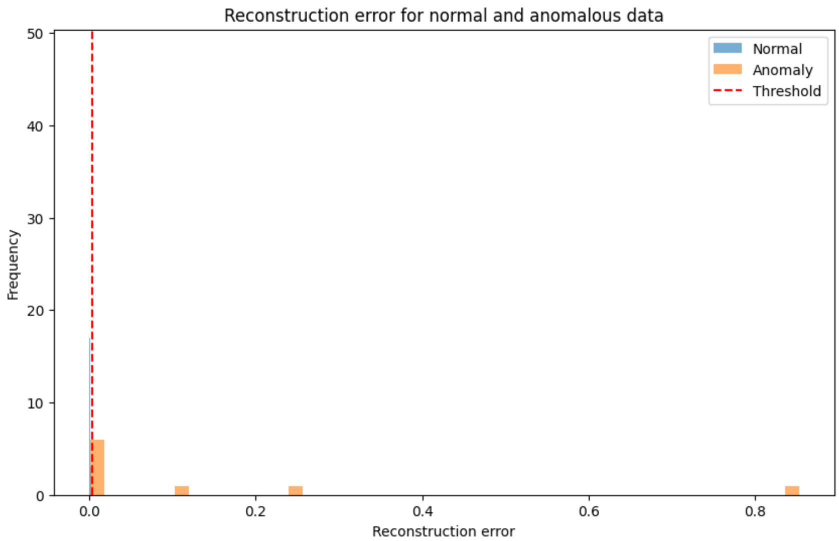
3.2 Anomaly Detection

Utilizing AutoEncoders for anomaly detection entails training the network with typical, representative data to acquire knowledge of its fundamental patterns and characteristics. After being trained, the AutoEncoder is able to reconstruct the input data and calculate the difference between the input and its reconstruction using a loss function like Mean Squared Error (MSE). Deviation from learnt patterns, known as anomalies, generally leads to increased reconstruction mistakes. AutoEncoders are highly successful at detecting anomalies in several areas, such as identifying fraudulent activities in financial transactions, monitoring equipment malfunctions in industrial environments, and detecting abnormal patterns in medical diagnostics. AutoEncoders offer a reliable approach to automatically identify and examine odd cases in huge datasets by measuring reconstruction errors. This capability improves proactive decision-making and enhances the reliability of the system.

The Notebook provides code for constructing a synthetic dataset that has a limited number of anomalies. The subsequent model attempts to detect and identify these anomalies. The synthetic dataset is displayed below:

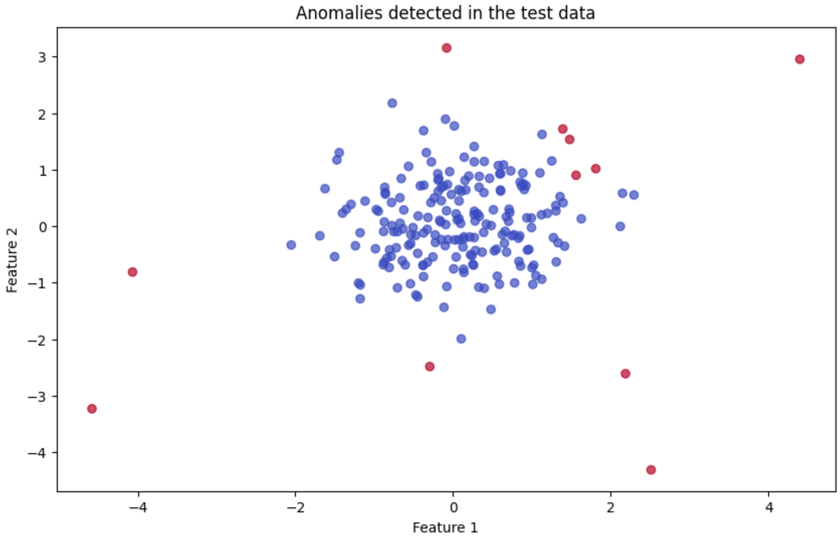


The AutoEncoder identifies anomalies by comparing the reconstruction error of input data with its output. Trained on normal data, it learns to reconstruct inputs accurately. During inference, inputs with significantly higher reconstruction errors are flagged as anomalies. This approach leverages the model's ability to encode normal patterns into a compressed representation and decode them back, making anomalies stand out due to their deviation from learned norms.

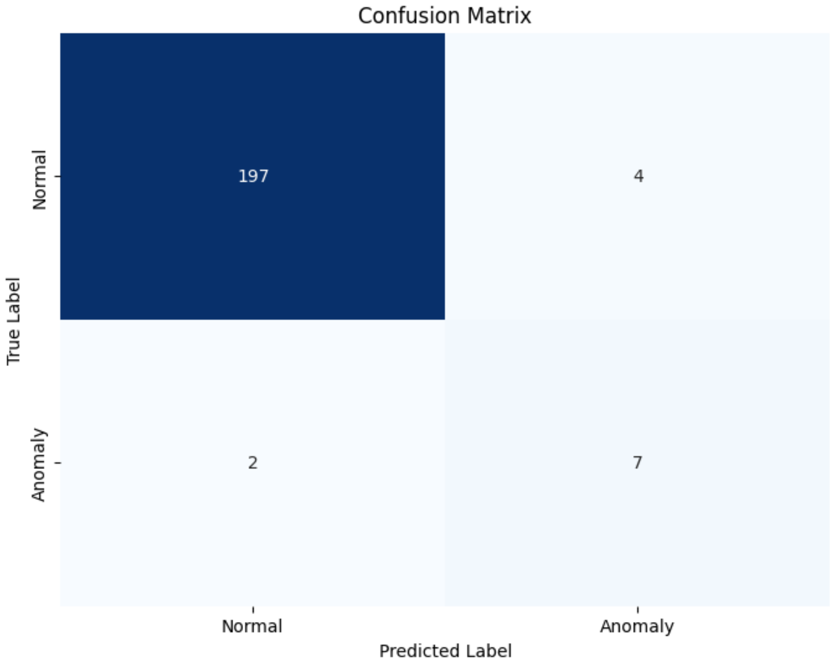


The image above shows the graph depicting the frequency of Reconstruction Error Values versus the value of the error. The frequency of the maximum Reconstruction error value indicates the number of anomalies identified in the training dataset by the model.

Results on the test data:



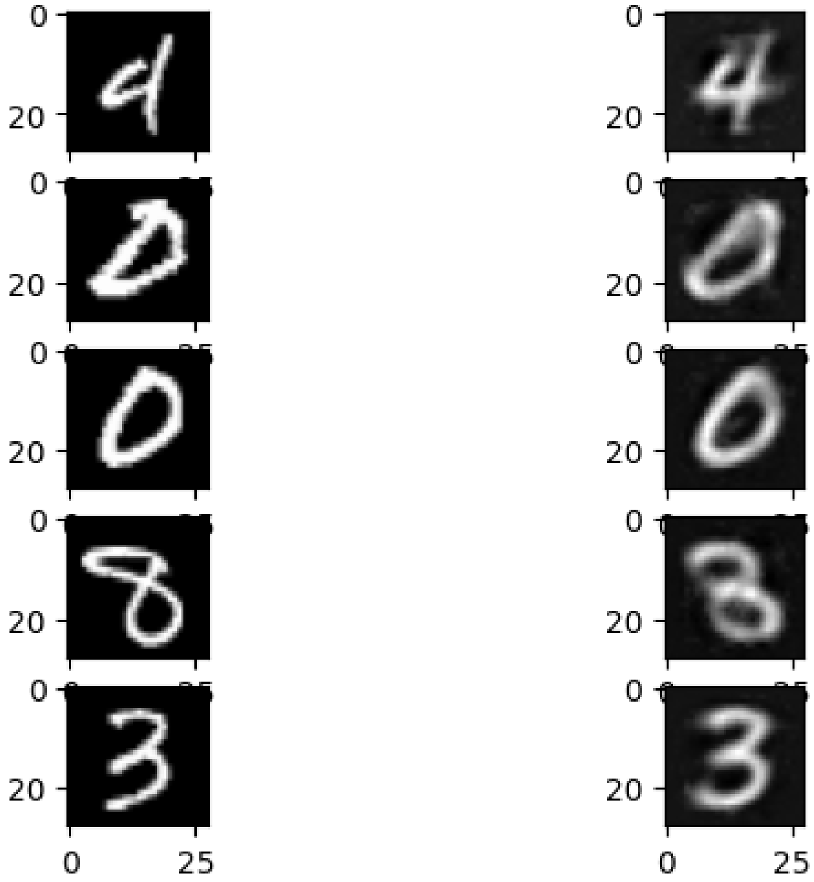
The confusion matrix:



3.3 Data Compression

The AutoEncoder utilizes a dense neural network to encode images into a 2-dimensional latent space, resulting in image compression. The compression is accomplished by decreasing the initial 784-dimensional image representation. Subsequently, the encoded data is deciphered back into the initial image dimensions by means of another neural network, while minimizing reconstruction error through the use of Mean Squared Error (MSE). This technique teaches the model to capture crucial characteristics while eliminating irrelevant details, enabling effective image representation and reconstruction.

The Results:



3.3.1 Why the images are not of high quality?

One factor contributing to the subpar quality is the utilization of a limited quantity of neurons (300) in the thick layer. Another rationale is the utilization of only two elements to represent all images. Enhancing the quality can be achieved by incorporating additional features, albeit at the expense of increasing the quantity of the compressed data.

Another rationale is the complete omission of convolutional layers. Dense layers are effective in capturing the overall characteristics of images, while convolutional layers excel in capturing the specific details and features. To improve the outcome, incorporating additional convolutional layers would be beneficial.

4 Conclusion

To summarize, AutoEncoders are effective tools in the field of machine learning, providing diverse applications such as removing noise from images, detecting anomalies, and compressing data. Although their success depends on parameters such as network architecture and latent space dimensions, they are extremely beneficial in extracting meaningful representations from complex data.